

# Configuring IP Spoofing on the Cache Engine in a Transparent Setup with a Content Services Switch

Document ID: 42162

## Contents

### Introduction

#### Before You Begin

- Conventions
- Prerequisites
- Components Used
- Background Theory

#### Configure

- Network Diagram
- Configurations

#### Verify

#### Troubleshoot

#### Related Information

## Introduction

This document provides a sample configuration for transparent caching and IP spoofing simultaneously without using Web Cache Communications Protocol (WCCP) on the Cisco Cache Engine and Cisco Content Services Switch (CSS) 11000 or CSS 11500 load balancer.

## Before You Begin

### Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

### Prerequisites

There are no specific prerequisites for this document.

### Components Used

The information in this document is based on the software and hardware versions below.

- Cache Engine (CE) 500 running Application and Content Networking Software (ACNS) 4.2 or later
- CSS 11000 or CSS 11500

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

## Background Theory

Transparent caching means that the traffic from a client to a server is silently redirected by a router or a Layer

4 switch to a cache device (Cisco Cache Engine in this case).

If the cache device already has a copy of the content the client is looking for, the cache will reply on behalf of the server. If the content is not present on the cache, the device will try to get it from the server before answering the client request.

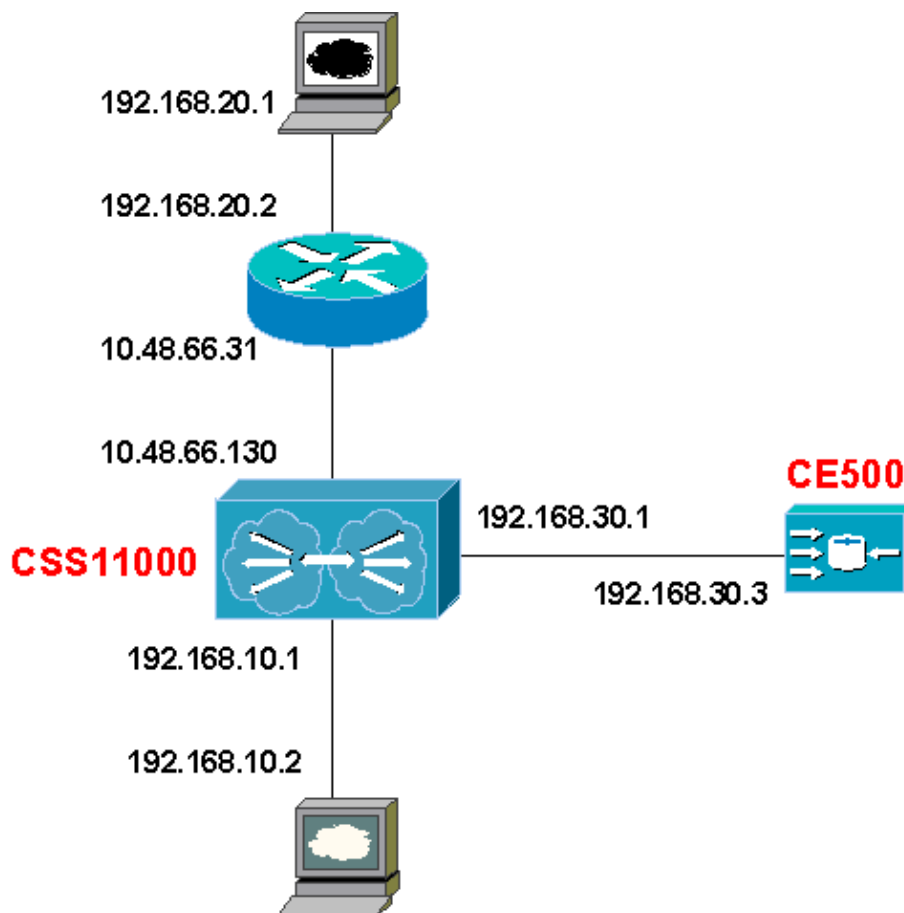
By default, the cache will contact the server using its own IP address. It is sometimes necessary, however, to use the client IP address. This is feasible by configuring IP spoofing.

## Configure

In this section, you are presented with the information to configure the features described in this document.

### Network Diagram

This document uses the network setup shown in the diagram below.



### Configurations

This document uses the configurations shown below.

- CSS 11000
- Cache Engine 500

**CSS 11000**

```
!Generated on 04/18/2003 09:30:41
!Active version: ap10500007s
```

```
configure
```

```
!***** GLOBAL *****
```

```
no restrict web-mgmt
no restrict xml
bridge spanning-tree disabled
persistence reset remap
acl enable
```

```
!--- An Access Control List (ACL) is needed. Enable the ACL.
```

```
ip route 0.0.0.0 0.0.0.0 10.48.66.1 1
ip route 192.168.10.0 255.255.255.0 192.168.20.100 1
ip route 192.168.20.0 255.255.255.0 10.48.66.31 1
ip route 192.168.20.0 255.255.255.0 192.168.30.3 1
```

```
!--- Very important !!!!
```

```
!--- For the ECMP feature of the CSS to work,
!--- you need one route pointing to the upstream router,
!--- and one identical route pointing to the cache.
!--- The CSS will know which one to use based on where
!--- the traffic came in first.
```

```
!***** INTERFACE *****
```

```
interface e1
  phy 100Mbps-FD
```

```
interface e2
  bridge vlan 149
  phy 100Mbps-FD
```

```
interface e3
  bridge vlan 161
  phy 100Mbps-FD
```

```
!***** CIRCUIT *****
```

```
circuit VLAN1

  ip address 10.48.66.130 255.255.254.0
```

```
circuit VLAN149

  ip address 192.168.10.70 255.255.255.0
```

```
circuit VLAN161

  ip address 192.168.30.1 255.255.255.0
```

```
!***** SERVICE *****
```

```
service agra
```

```
!--- Definition of the cache device.
```

```
ip address 192.168.30.3
type transparent-cache
```

```
!--- It is important to set the type to transparent-cache
!--- so that the CSS does not NAT the destination IP address.
!--- Only the destination MAC address is modified.
```

```

port 80
active
!***** EQL *****
eql CacheMe

!--- Definition of what objects are cacheable.

extension gif
extension html
extension pdf
extension zip
extension gz

!***** OWNER *****
owner gilles
content ToCache

!--- Definition of the content rule to redirect the traffic.
!--- No VIP address specified since you want to intercept all HTTP traffic.

protocol tcp
port 80
url "/*" eql CacheMe

!--- Redirect all requests of a cacheable object.

add service agra
active
!***** ACL *****
acl 1
clause 10 bypass tcp any destination 192.168.10.2 eq 80

!--- This ACL is necessary to make sure that the HTTP requests from
!--- the cache itself are not intercepted by the content rule.

clause 20 permit any any destination any
apply circuit-(VLAN161)
acl 2

!--- Permit all traffic for the other interfaces.

clause 20 permit any any destination any
apply circuit-(VLAN149)
apply circuit-(VLAN1)

```

## Cache Engine 500

```

hostname CE500
!
http 14-switch enable

!--- Tells the Cache Engine to accept traffic with any IP destination.

http 14-switch spoof-client-ip enable

!--- This is a new command in ACNS 5.x. this command replaces the
!--- wccp spoof-client-ip enable command.

```

```

!
!
!
!
!
!
exec-timeout 0
!
!
!
interface FastEthernet 0/0
 ip address 192.168.30.3 255.255.255.0
 exit
interface FastEthernet 0/1
 shutdown
 exit
!
!
ip default-gateway 192.168.30.1
!
primary-interface FastEthernet 0/0
!
!
!
logging console enable
!
!
!
!
!
!
wccp version 2
wccp spoof-client-ip enable

!--- This commands enable IP spoofing, and it works
!--- even if you do not use WCCP. This command only works with
!--- WCCP redirected traffic if you have ACNS 5.x.
!--- Therefore, if you are using version 5.x of ACNS, this command
!--- should be replaced with the command http 14-switch spoof-client-ip enable
!--- mentioned above.

!
!
CE500#

```

## Verify

This section provides information you can use to confirm your configuration is working properly.

Certain **show** commands are supported by the Output Interpreter Tool (registered customers only) , which allows you to view an analysis of **show** command output.

### CSS 11000 show Commands

- **show summary** Shows the content rule hit counters to see if CSS is receiving and redirecting traffic.
- **show service *name*** Shows the status of the service.

## Cache Engine Commands

- **show stat http request** *name* Shows the number of HTTP request received by the cache.
- **show stat http savings** *name* Shows the number of hits and miss on the Cache Engine.

## Troubleshoot

To troubleshoot this problem, use the commands above. Most of the time, however, it is necessary to use a sniffer in order to find out the exact path followed by the traffic.

You can also issue the CSS **flow trace** command available in debug mode. You will get better result with a sniffer however.

## Related Information

- **Content Networking Downloads** ( registered customers only)
- **Content Networking Devices Hardware Support**
- **Technical Support – Cisco Systems**

---

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2013 – 2014 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Sep 01, 2004

Document ID: 42162

---